IN THE SPECIFICATION

Please amend the portions of the Specification identified below to read as indicated herein.

Paragraph beginning at page 2, line 19:

Figure 1 shows a block diagram of a generalized search engine 10. User terminal 15, text engine 20, database 35, and sorting ~~processor~~ process 65 are all connected through network 40.

Paragraph beginning at page 2, line 22:

User terminal 15 is typically capable of generating a query, receiving and displaying the results of that query, and retrieving and displaying documents included in the results. User terminal 15 may be operated by a person or may generate queries in response to a program or an automated process. For purposes of the invention, a user may include a person, program, automated process, or any other device or technique for generating queries for a search engine. Text engine 20 includes capabilities for directing the addition of documents 50 to database 35, and initiating index processes ~~60~~ 75, search processes 25, and intersection processes 30. Text engine 20 also includes capabilities for initiating a process 45 for assigning unique identifiers 70 to documents, and for generally controlling the activities of search engine 10. Documents 50 and index files 55 are typically located in database 35.

Paragraph beginning at page 3, line 4:

Documents 50 may be loaded into database 35 either manually or automatically under the direction of text engine 20. As part of the loading process, text engine 20 may first assign a random number to each document as a file name or document key, also known as a unique identifier 70, through unique identifier process 45. Text engine 20 may also initiate indexing processes ~~60~~ 75 that generate and update various index files 55. Indexing files 55 may include a table of unique words identified in each document 50. In addition, for each word in the unique words table, indexing processes ~~60~~ 75 may add pointers to the table pointing to the documents containing that word. Indexing processes ~~60~~ 75 may also create other index files 55 including ones containing the number of occurrences of each word in each document and their location within each document.

Paragraph beginning at page 3, line 14:

Once database 35 is operational, a user may generate a query using user terminal 15. The query usually includes a number of key words which may be connected by logical operators (e.g., AND, OR, NOT, etc.) The query is submitted to text engine 20 which initiates at least one search process 25. For complex queries, text engine 20 may initiate a number of search processes 25, one for each component or segment of the query. If a single search process 25 is utilized, the search process 25 will return a list of documents that satisfy the search criteria. ~~A sorting~~ Sorting process 65 will typically sort the list in unique identifier order. The items in the list may be given a rank as to relevance and then displayed on user terminal 15. In the case where multiple search processes 25 are employed, when the search processes 25 are complete, text engine 20 coordinates at least one intersection process 30 that generates a list of documents that are common to each of the search results. The list is then sorted in unique identifier order by sorting process 65. The document list may then be ordered according to relevancy and then presented to the user through user terminal 15. Multiple search processes 25 and intersection processes 30 typically take significant processing time to complete and also consume relatively large areas of storage space. This may introduce delays and storage management problems if the intermediate results from the individual search processes 25 are large.

Paragraph beginning at page 4, line23:

This invention is directed to a device for retrieving stored ~~data~~ data. The device includes means for assigning at least one prioritized attribute to the data prior to storage and means for retrieving the stored data, where the stored data is retrieved in an order determined by the priority of the at least one prioritized attribute assigned to the stored data. The stored data may include an identifier, and the at least one prioritized attribute may be encoded into the identifier. The stored data, means for assigning, and means for retrieving may be connected to and distributed over a network having a plurality of nodes.

Paragraph beginning at page 5, line 16:

Figure 2 shows an example of a computing device 200 embodied as a unique search engine in accordance with the teachings of the invention. User terminal 210, text engine 215, database 220, and sorting ~~processor~~ process 225 are all coupled to network 230.

Paragraph beginning at page 5, line 24:

Database 220 typically includes index files 255 and documents 260. Sorting ~~processor~~ process 225 operates on the results of a search process 240 when a single search process 240 has been initiated, and sorts the results in document key order. When multiple search processes 240 are initiated and intersection process 245 is used to intersect the results of the search processes 240, sorting ~~processor~~ process 225 sorts the results of the intersection process 245 by document keys. In either case, the sorted list of documents may be displayed to the user through user terminal 210. If the user is a program or process, the sorted list of documents may simply be passed to the program or process.

Paragraph beginning at page 6, line 7:

Text engine 215 directs the loading of documents 260 into database 220. According to the invention, as part of the loading process, text engine ~~220~~ 215 assigns a special document key 265 to each document utilizing unique identifier process 250. Special document key 265 can begin as a random number, or any other document identifier that may be initially generated by text engine 215. In addition, unique identifier process 250 encodes one or more document attributes into the special document key 265, thus producing a unique identifier that includes certain attributes of the document 260. Examples of attributes that may be encoded in special document key 265 may include the date the document was created, the size of the document, the number of occurrences of a specific word or words, or any other attributes of the document 260 that are suitable for encoding. The document 260 with its special document key 265 is then stored in database 220. As part of the loading process text engine 215 may also initiate various indexing processes 235 that create any number and type of index files 255 in database 220.

Paragraph beginning at page 7, line 1:

Figure 4 shows the operation of computing device 200 utilizing special document key 265. A user generates a query which is submitted to text engine 215 in step 400. In step 410 text engine 215 initiates a search process 240 based on the query. In step 420, the search process retrieves a list of documents 260 that satisfy the search criteria. Sorting ~~processor~~ process 225 then sorts the list in document key order in step 430.

Paragraph beginning at page 7, line 6:

In a preferred embodiment, unique identifier process 250 encodes attributes in special document key 265 such that sorting ~~processor~~ process 225, in sorting the list of documents in document key order, actually sorts the document list in attribute order. In other words, special document key 265 is constructed so that the attributes are represented in a specific manner in special document key 265, such that when sorting ~~processor~~ process 225 sorts the retrieved list by document keys, it also sorts the retrieved list in attribute order. Thus, as shown in step 440 of Figure 4, sorting ~~processor~~ process 225 yields a list in attribute order.

Paragraph beginning at page 7, line 14:

This is advantageous in that, if a user knows how the attributes are encoded in the special document key 265, or at least how the attributes will be ordered by sorting ~~processor~~ process 225, the user may construct queries that require a minimum number of multiple search processes 240 and avoid intersection processes 245. Utilizing these queries, text engine 215 may return a document list already sorted in order of the attributes desired by the user. In addition, the document list is produced in a reduced time period and with less of an impact on system resources than conventional searching techniques. Also, by understanding how the attributes will be ordered, a user has the ability to construct queries that yield results that are organized in a manner that is more useful to the user and that include an increased number of relevant documents.

Paragraph beginning at page 8, line 5:

Unique identifier process 250 encodes the published date attribute in special document key 270 such that sorting ~~processor~~ process 225 will sort a list of documents returned from search process 240 or intersection process 245 in published date order.

Paragraph beginning at page 8, line 8:

The user generates a query for documents having a specific word combination which is submitted to text engine 215. A search process 240, initiated by text engine 215 returns a list of documents satisfying the query. When sorting process 225 sorts the results of the search process 240, the sorted document list includes all documents having the specific word combination in published date order. Thus, multiple search processes 240 have been minimized and the intersection process 245 has been avoided by coding a particular attribute into the special document key 270.